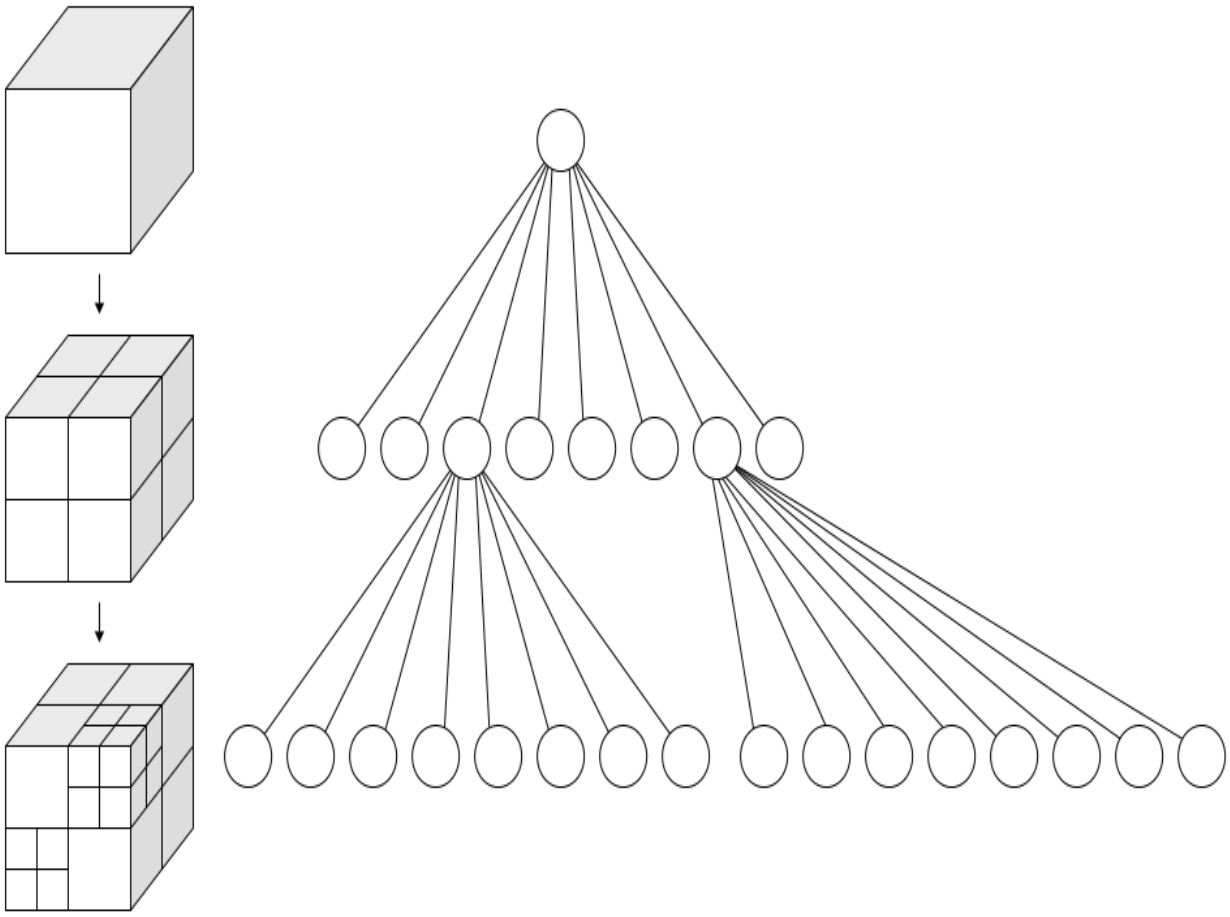
Tema: Octree

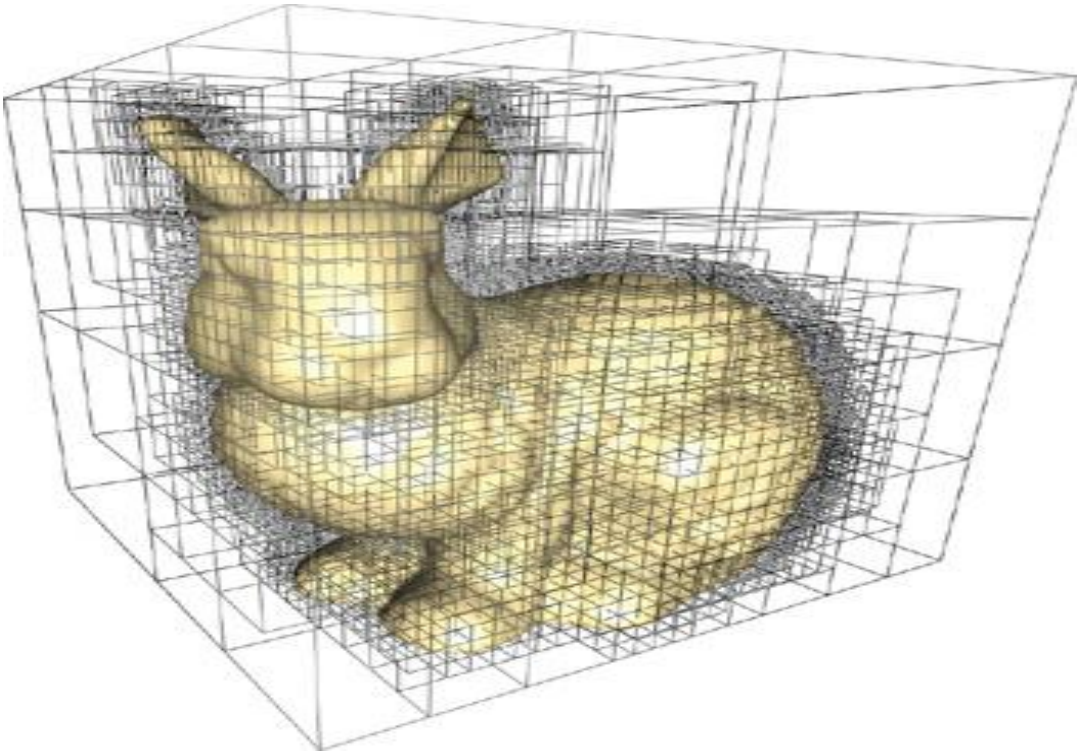
Numele aplicatiei: Paint

1)Aspecte teoretice

Octree este o structura de date tip arbore cu fiacre nod intern avand exact opt copii. Octree sunt cel mai des folositi in impartirea spatiului tridimensional folosind diviziuni recursive in opt octanti.



Este folosit adesea in grafica 3D si la motorele jocurilor 3D



2)Aspecte generale ale aplicatiei

Aplicatia reprezinta o variant clasica a paint-ului din Windows fiind adaptata pentru a putea folosi octree la creearea acesteia.

Caracteristici:

1. Aplicatia confera posibilitatea de a schimba culoarea patratelelor ce se vor selecta prin intermediul mouse-ului.
2. Aplicatia contine 3 tipuri de pensule:
   a) Pensula mica
   b) Pensula medie (de 8 ori mai mare ca cea initiala)
   c) Pensula mare (de 8 ori mai mare ca cea medie)
3. Aplicatia este simpla si usor de utilizat

3)Aspecte tehnice ale aplicatiei.Implementare

Limbajul folosit pentru realizarea aplicatiei este C++ la care se adauga librarii de grafica ce folosesc standardul OpenGL (GLU, GLEW, SDL).

Implementarea panzei se bazeaza pe impartirea acestea folosind metoda octree. Spre exemplu plansa mica este impartita in 64 de patrate.

OpenGL (Open Graphics Library) este o specificație a unui standard care definește un API (Application Programming Interface) multiplatformă foarte utilizat pentru programarea componentelor grafice 2D și 3D ale programelor de calculator. Interfața constă în peste 250 de apeluri diferite care folosesc la a desena pe ecranul calculatorului scene 3D complexe din primitive (din primitives, elemente simple).

Un **shader** este o funcționalitate executată pe procesorul grafic, care redă o parte din scena 3D în aplicații grafice.

Shaderii sunt scrisi intr-un limbj de prigramare asemanator C-ului numin GLSL.

Sunt 2 tipuri de shadere: Vertex shader și Fragment/Pixel shader. De obicei ele se utilizează în tandem. Vertex shader se execută pentru fiecare vector din modelele afișate. După aceasta modelele se rasterizează, și pentru fiecare pixel individual după aceasta se execută Pixel/Fragment shader. Un vertex shader poate determina culorile din jurul acelui vector, poate schimba poziția sa în spațiu.

Un pixel shader de obicei calculează culoarea și transparența pixelului respectiv. Pixel shader se execută practic înainte de afișare, când deja majoritatea lucrului cu poziționarea este efectuat.

Din cauza că există mult mai mulți pixeli decât vectori, execuția lor durează mai mult timp decât procesarea vectorilor.

Exemplu de Vertex Shader

```
#version 400

attribute vec3 position;
uniform mat4 transform;


void main()
{

    gl_Position = transform * vec4(position, 1.0);

}
```

Exemplu de fragment Shader

```
#version 400

uniform vec4 new_color;
out      vec4 output_color;

void main()
{
    output_color = new_color;
}
```

Proiectanti:

Modrogeanu Cosmin-Ionut

Iosa Marius-Alexandru